C语言的秘密

——汇编语言大作业

班级： 1403201

姓名： 唐帮盈

学号：1140320116

**C语言的语言元素**：

**数据类型**：

1 整形（long int，int，short）；

2 浮点型（float，double）；

3 字符型（char）；

4数组：基于各种数据类型进行的定义。

**运算符和宏定义**：

1宏定义（#define NUM 20）；

2算数运算符进行计算，逻辑运算符进行逻辑运算，位运算符进行位的运算，包括（‘+’，‘-’，‘\*’，‘/’，‘%’等算数运算符，‘&&’，’||‘，‘！’等逻辑运算符，‘&’，‘|’等位运算符）

**子程序：**

1形参，实参，局部变量；

2子程序的相互调用，进而进行堆栈压栈出栈。

**标识符：**

1可供调用的库函数，如strlen，strcmp，strcpy；

2关键字，如struct，union等；

3单目操作符，如sizeof。

**汇编语言中：**

**数据类型：**

1分为常用的8位，16位，32位的整数，分别为(s)byte,(s)word,(s)dword；并还有fword（48位），qword（64位），tbyte（80位）等整数；

2实型：real4（4字节），real8（8字节），real10（10字节）；

3数组，与C语言类似，数组名代表的是第一个元素的地址；

运算：

1汇编语言使用机器指令（不支持两个数直接使用运算符进行计算），所以就有了一系列的助记符，比如（add，加，sub，减，mul，imul，乘，xchg，交换值，mov，赋值等等）；

2以及位运算都是使用助记符实现（shl，shr，sal，sar，rol，rcr，rcl，ror等等）。

**子程序（进程）：**

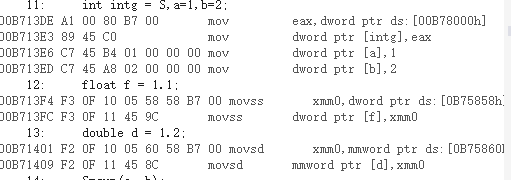
1局部变量的使用；

2对于堆栈的处理

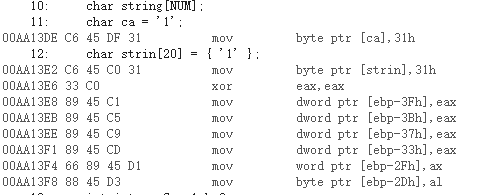
自然，作为机器指令的对应，一条C语言对应着多条的汇编语言，其数据类型也有着相应的对应关系。主要通过反汇编来进行研究。

Int 对应的是dword

Float 对应的是dword

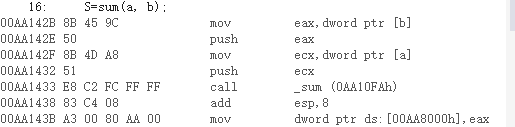
Double 对应的是mmword

char 则与byte对应

sting 即char[] 则与byte数组对应

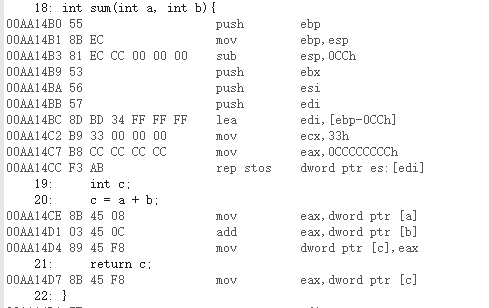
至于子程序，在汇编语言里称为进程

则是通过对于堆栈的操作，将之前的地址与值进行压栈保存，从而实现了调用函数，并在函数中使用局部变量，



这段代码是关于调用一个求和函数时进行的一系列操作，并且实现了调用者释放堆栈空间，将实参整数a和实参整数b的值压栈，从而使得函数可以进行调用。

至于函数的声明与使用：

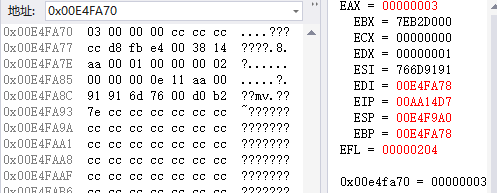


这是一个简单的求和代码的反汇编的代码，分别有形参和局部变量，首先还是将ebp压栈，将esp的值赋给ebp，即把堆栈里现在指向的指针存至ebp，从而能够通过ebp对堆栈内存的值进行调用。

接下来对这个进程进行分析进而对于C语言的子程序的实现进行讲解：

ESP与ebp的处理，参考上一段；

将esp减去0cch从而在堆栈中申请了0cch字节的内存，这一段的内存供该进程使用（主要是进行申请局部变量使用），如图中所示，局部变量c的地址为00e4fa70h，而ebp中值为00e4fa78h，是ebp-8；



返回值的问题，C语言的返回都是通过eax进行的，所以在最后一步进行了将所要

返回的值赋值给了eax。

程序的形参问题，子程序的形式参量存储在堆栈中，就是在ebp之上（ebp+8）。

接下来分析两者的全局变量，C语言的全局变量应该就是直接在代码段进行了声明，而

有了全局变量并且每个进程都能使用。所以二者的全局变量使用应该都是一样的。

**C语言和和汇编语言孰优孰劣？**

相对于python，C语言的优势在于比之更低级，换言之，更接近于机器指令，那么，在进行编译的时候就需要更少的机器指令，从而能够提高程序的执行效率。同样的，一句C语言指令往往对应着多条的汇编指令即对应着多条的机器指令。我们可以看，每次C语言的子程序进行使用的时候，要申请一大段的堆栈空间，但是如果是我们进行编程而不是编译器生成，自然能够节省很大的一部分空间，并且在一些时候使用更高效语句进行描述，从而提高程序的执行效率。但是，因为汇编程序的代码的复杂和难记，很少有人愿意去使用汇编程序编辑一个很大的程序。但是在处理一些细节方面，汇编语言有着其他语言难以比拟的优势，所以，在进行一些编程的接口的实现的时候我们可以使用汇编语言来进行提高程序的执行效率。总的来说，并没有哪种语言是真的所有的地方都是好的，只能够说是他们在某个方面有着其他的语言所无法比拟的优势，所以在合适的地方采取最合适的编程语言才是真的最好的地方。

我认为的C语言适用的范围，主要进行程序的内核开发时，进行算法描述时，因为其相对于那些高级语言来说，更底层，更能够提高程序的运行效。而汇编语言更适合在编程时做为一个辅助，在一些接口的时候采用汇编语言提高了其程序的效率。虽然可能麻烦了一点，但是却提高了程序的执行效率并减少了对于空间的占用。

**学习汇编语言的体会：**

刚开始的时候是一种好奇，并且在老师展示了程序可以不需要数据库就直接通过控制接口就使计算机发出了声音，引起了我的兴趣。真的感觉挺有意思的，真的想把这种语言掌握好。所以，课上的东西，不敢说都会，但当时听的时候真的是听懂了。

之后，讲到了高级进程的时候，可能有上课跑神儿的原因，有点儿听不懂了，并且有的伪指令需要ret，有的又不需要，那个时候真的是有点儿迷糊了虽然课本上就有可是因为没有认真的复习，从而感觉对这一段有些力不从心。这个感觉是，汇编好乱，好杂。

再后，学习16位编程和BIOS设计的时候，感觉真的有点儿上课听懂了，然后下课没用就忘了。所以就有了，不会的感觉。

**收获：**

**学习汇编语言收获挺大的，首先在最初初步了解了计算机的底层结构，加深了我对于计算机的认识；在学习补码等编码方式时，更进一步对于计算机的存储有了了解。**

**在写汇编程序的时候明白了一些高级程序的实现过程，进而能够提高自己的程序效率（使用哪些语句有更高的效率）。**

**并且对于堆栈的使用也有了进一步的认识。**

**还有，子程序的调用，在C语言中直接进行调用就可以，但是在汇编语言中却需要一系列的操作，才能实现。**

**对于编程使用价值最高的恐怕是混合编程了，这样能够让我在C语言中适当的调用汇编语言，从而提高程序的执行效率。**